

Word sense disambiguation via bipartite representation of complex networks

Edilson A. Correa Jr., Alneu de Andrade Lopes and Diego R. Amancio

*Institute of Mathematics and Computer Science
University of São Paulo (USP)
São Carlos, São Paulo, Brazil*

Abstract

The word sense disambiguation (WSD) task aims at identifying the meaning of words in a given context for specific words conveying multiple meanings. This task plays a prominent role in a myriad of real world applications, such as machine translation, word processing and information retrieval. Recently, concepts and methods of complex networks have been employed to tackle this task by representing words as nodes, which are connected if they are semantically similar. Despite the increasingly number of studies carried out with such models, most of them use networks just to represent the data, while the pattern recognition performed on the attribute space is performed using traditional learning techniques. In other words, the structural relationship between words have not been explicitly used in the pattern recognition process. In addition, only a few investigations have probed the suitability of representations based on bipartite networks and graphs (bigraphs) for the problem, as many approaches consider all possible links between words. In this context, we assess the relevance of a bipartite network model representing both feature words (i.e. the words characterizing the context) and target (ambiguous) words to solve ambiguities in written texts. Here, we focus on the semantical relationships between these two type of words, disregarding the relationships between feature words. In special, the proposed method not only serves to represent texts as graphs, but also constructs a structure on which the discrimination of senses is accomplished. Our results revealed that the proposed learning algorithm in such bipartite networks provides excellent results mostly when *topical* features are employed to characterize the context. Surprisingly, our method even outperformed the

support vector machine algorithm in particular cases, with the advantage of being robust even if a small training dataset is available. Taken together, the results obtained here show that the proposed representation/classification method might be useful to improve the semantical characterization of written texts.

Keywords: complex networks, bipartite graphs, word sense disambiguation, bipartite networks, pattern recognition, semantic analysis, network science

1. Introduction

The word sense disambiguation (WSD) task has been widely studied in the field of Natural Language Processing (NLP) [1]. This task is defined as the ability to computationally detect which sense is being conveyed in a particular context [2]. Although humans solve ambiguities in an effortlessly manner, this matter remains an open problem in computer science, owing to the complexity associated with the representation of human knowledge in computer-based systems [3]. The importance of the WSD task stems from its essential role in a variety of real world applications, such as machine translation [4], word processing [5], information retrieval and extraction [6, 7, 8, 9, 10, 11]. In addition, the resolution of ambiguities plays a pivotal role in the development of the so-called semantic web [12].

Many approaches devised to solve ambiguities in texts employ machine learning methods to automatically extract the best features in specific contexts [2]. Automatic methods commonly use texts as a source of information, and these texts need to be transformed into a structured format. Popular representations are vectors of features, trees and graphs of relations between words [1]. All such representations attempt to grasp, in a particular way, the semantical features related to the context surrounding ambiguous (target) words. Then, the information extracted from the context is used in the learning process. Although graphs have been employed in general pattern recognition methods [13, 14] and, particularly in the analysis of the semantical properties of texts in several ways [15, 16, 17, 18, 19, 20, 21], the use of network models in the learning process has been restricted to a few works (see e.g. [22, 23]). In addition, most of the current network models emphasise the relationship between *all* words of the document. As a consequence, a minor relevance has been given

to the relationships between feature and target words. In this paper, we propose a different network representation which does not consider the relationship between all words, as described e.g. in [17, 22]. We rather model texts using a bipartite network representation which focus on the relevant information arising from the relationship between *feature* and *target* words. This representation is then used as an underlying structure on which the proposed learning algorithm is applied. As we shall show, the combination of this textual representation and the proposed learning technique may improve the classification process when compared with well-known supervised algorithms hinging on traditional text representations. Remarkably, we have also found that our method retains its discriminative power even when a considerable small amount of training instances is available.

The remainder of this paper is organized as follows. Section 2 presents a brief review of basic concepts employed in this paper and related works. Section 3 presents the details of the proposed representation and algorithm to undertake the word sense disambiguation task. In Section 4, we discuss the details of the experiments and the results concerning the accuracy and robustness of the proposed method. Finally, we present some perspectives for further works.

2. Related works

The word sense disambiguation task can be defined as follows. Given a document represented as a sequence of words $T = \{w_1, w_2, \dots, w_n\}$, the objective is to assign appropriate sense(s) to all or some of the words $w_i \in T$. In other words, the objective is to find a mapping A from words to senses, such that $A(w_i) \subseteq \mathcal{S}_D(w_i)$, where $\mathcal{S}_D(w_i)$ is the set of senses encoded in a dictionary D for the word w_i , and $A(w_i)$ is the subset of appropriate senses of $w_i \in T$. One of the most popular approaches to tackle the WSD problem is the use of machine learning, since this task can be seen as a supervised classification problem, where senses represent the classes [2]. The attributes used in the learning methods are usually any informative evidence obtained from the topical context and external knowledge sources. The latter approach is usually not common in practice because the creation of

knowledge datasets demands a time-consuming effort, since the change in domains requires the recreation of new knowledge bases.

The generic WSD task can be distinguished into two types: *lexical sample* and *all-words* disambiguation. In the former, a WSD system is required to disambiguate a restricted set of target words. This is mostly done by supervised classifiers [2]. In the *all-words* scenario, the WSD system is expected to disambiguate all open-class words in a text. This task usually requires a wide-coverage of domains, and for this reason a knowledge-based system is usually employed. In this article, only the *lexical sample* task is considered.

The main step in any supervised WSD system is the representation of the context in which target words occur. The set of features employed typically are chosen to characterize the context in a myriad of forms [2]. The most common types of attributes used for this aim are:

- *local features*: the features of an ambiguous concept are a small number of words surrounding target words. The number of words representing the context is defined in terms of the window size ω . For example, if the context of the target word τ_ω is “ $p_{-3} \ p_{-2} \ p_{-1} \ \tau_\omega \ p_{+1} \ p_{+2} \ p_{+3}$ ” and $\omega = 2$, then the words p_{-2} , p_{-1} , p_{+1} and p_{+2} are used as features.
- *topical features*: the features are defined as topics of a text or discourse, usually denoted in a bag-of-words representation;
- *syntactical features*: the features are syntactic cues and argument-head relations between the target word and other words within the same sentence; and
- *semantical features*: the features of a word are any semantic information available, such as previously established senses or domain indicators.

Using the aforementioned set of features, each word occurrence can be converted to a feature vector, which in turn is used as input in supervised classification algorithms. Typical classifiers employed for this task include decision trees [24], bayesian classifiers [24, 25], neural networks [24] and support vector machines [25, 26].

Another approach that has been used to address the WSD problem consists in the use of complex networks and graphs [16]. For instance, the HyperLex algorithm [17] connects words co-occurring in paragraphs to establish similarity relations among words appearing in the same context. The frequency of co-occurrences is considered according to the following weighting scheme:

$$w_{ij} = 1 - \max\{P(w_i, w_j), P(w_j, w_i)\} \quad (1)$$

where $P(w_i, w_j) = f_{ij}/f_i$, f_i is the frequency of word i in the document and f_{ij} is the frequency of the co-occurrence of the words i and j . Then, this network is used to create a tree-like structure via recognition of central concepts, which represent all possible senses. To perform the classification, the distance of context words to the central concepts in the tree structure is computed to identify the most likely sense.

Using a different approach, [15] uses the local topological properties of co-occurrence networks to disambiguate target words. In this case, even though a significant performance has been found for particular target words, the optimal discrimination rate was obtained with traditional local features, suggesting thus that the overall discriminability could be improved upon combining features of distinct nature, as suggested by similar approaches [27, 28, 29].

Despite the numerous studies devoted to the WSD problem, this task remains an open problem in NLP, and currently it is considered one of the most complex problems in Artificial Intelligence [3]. Our contribution in this paper is the proposition of a new representation that is able to focus the sense discrimination analysis on the relationship between features and target words. Unlike previous studies [15, 17], we disregard the links between features words in our bipartite graph representation. Despite its seemingly simplicity, we show that such representation captures, in a artlessly manner, informative properties of target words and their respective senses.

3. Overview of the technique

This section presents the approaches to represent local and topic features of target words in a bipartite heterogeneous network. Here we also present the Inductive Model Based on

Bipartite Heterogeneous Network (IMBHN) algorithm, which is responsible for inducing a classification model from the structure of a bipartite network [30, 31].

3.1. Modelling word context as a bipartite heterogeneous network

Traditionally, the context of ambiguous words is represented in a vector space model, so that each target word is characterized by a vector. In this representation, each dimension of the vector corresponds to a specific feature. Alternatively, we may represent the data using a bipartite heterogeneous network. In this model, while the first layer comprises only feature words, the second only stores target words. As mentioned in Section 2, currently, there exists a wide variety of features to tackle the WSD problem. In this paper, we focused on the analysis of *local* and *topical* attributes, as such data are readily available on (or derivable from) any corpus. Note that, in this case, we have not used any knowledge dataset.

In the proposed strategy based on *topical* features, we create a set \mathcal{T} of topical words. Then, each distinct becomes a distinct feature. As topical words, we considered the most frequent words of the dataset. The number of topical words, i.e. $|\mathcal{T}|$, is a free parameter. Given \mathcal{T} , the bipartite network is created by establishing a link between topical and target words whenever they co-occur in the same document.

In the proposed representation based on *local* features, each feature word surrounding the target word represents an attribute. For each instance of the target word in the text, we select the ω closest surroundings words to become a feature word (see definition in Section 2). The selected words are then connected to the target words by weighted edges.

3.2. Algorithm description

The IMBHN algorithm can be used in the context of any text classification task. If the objective is to classify distinct documents in a given number of classes, the bipartite network can be constructed so that nodes represent both terms and documents. In this general scenario, such representation is used to compute the relevance of specific terms for distinct document classes. In a similar fashion, in this study, we compute the relevance of *local/topical* features for each target word. Then, this relevance is used to infer word senses.

The proposed algorithm for sense identification relies upon a network structure with two distinct layers: (i) a layer representing possible feature words (i.e. *local* or *topical* features), and (ii) a layer comprising all occurrences of the target word. The two layers are illustrated in Figure 1. Edges are established across layers so that context words and distinct occurrences of the target word are connected. In addition, in the proposed network representation, a weight relating each feature word to each target word is also established. The main components of the model are:

- w_{d_k, t_i} : the weight of the connection linking the k -th target word and the i -th feature word. In the strategy based on *topical* features, this weight is constant along the execution of the algorithm and, for a given document T , is computed as

$$w_{d_k, t_i} = 1 - \delta(d_k, t_i)/l(T), \quad (2)$$

where $\delta(d_k, t_i)$ denotes the the distance between two words (i.e. the number of intermediary words) and $l(T)$ is the length of T (measured in terms of word counts). In the strategy based on *local* features, the weight of the links is given by the term frequency - inverse document (tf-idf) strategy [1].

- f_{t_i, c_j} : let \mathcal{C} be the set of possible classes (i.e. word senses). f_{t_i, c_j} represents the current relevance of the i -th feature word ($t_i \in \mathcal{T}$) to the j -th class ($c_j \in \mathcal{C}$). This value is initialized using a heuristic and then is updated at each step of the algorithm.
- y_{d_k, c_j} : represents the *actual* membership of the k -th target word. In other words, this is the label provided in the supervised classification scheme. If c_j is the class of the k -th target word, then $y_{d_k, c_j} = 1$; otherwise, $y_{d_k, c_j} = 0$.
- ϕ_{d_k, c_j} : represents the *obtained* membership of the k -th target word. If c_j is the class obtained for the k -th target word, then $\phi_{d_k, c_j} = 1$; otherwise, $\phi_{d_k, c_j} = 0$.
- ϵ_{d_k, c_j} : denotes the error of the current iteration. It is computed as:

$$\epsilon_{d_k, c_j} = y_{d_k, c_j} - \phi_{d_k, c_j}. \quad (3)$$

As we shall show, this error is used to update weights in f so that, at each new iteration, the distance between y_{d_k, c_j} and ϕ_{d_k, c_j} decreases.

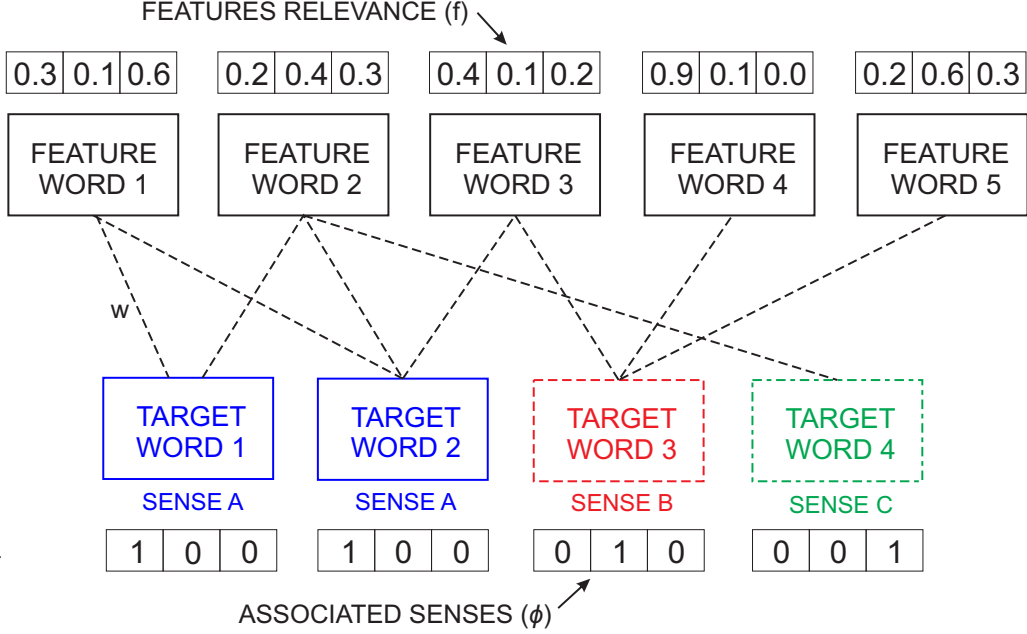


Figure 1: Bipartite network structure used by the IMBHN algorithm. Note the existence of two layers: the layer comprising feature words and the layer comprising target words, which can be classified into three distinct senses (A, B and C). For each feature word, there exists a vector of features relevance whose element f_{t_i, c_j} denotes the relevance of i -th feature word for the j -th possible sense. The vectors below each target word represents the sense obtained in each iteration (i.e. ϕ_{d_k, c_j}).

Note that, in the model illustrated in Figure 1, we only consider the relationship between feature and target words. Differently from traditional models, the relationship between feature words [15] is not explicitly considered in our model.

The training phase of the algorithm can be divided into the three following major steps:

1. **Initialization:** there are three possible ways of initializing f , i.e. the vector weights of feature words. The most simple strategy is to initialize weights with zeros or random values. A more informed alternative initializes weights using the a priori likelihood of feature words co-occur with senses. This probability can be computed as

$$\Pr = P(f_i | d_k) = n_{f_i, d_k} / n_{d_k}, \quad (4)$$

where n_{f_i, d_k} is the number of times that the i -th feature word appears in the context of the k -th target word and n_{d_k} is the total number of occurrences of d_k . In our experiments, we report the best results obtained among these three alternatives.

2. **Error calculation:** In the error calculation step, firstly, the output vector for each target word ($\phi(d_k)$) is computed. This vector depends upon the presence of the feature word in the context (w_{d_k, t_i}) and its relevance for the class (f_{t_i, c_j}). Mathematically, the class computed at each new iteration is given by

$$C\left(\sum_{t_i \in \mathcal{T}} w_{d_k, t_i} f_{t_i, c_j}\right) = \begin{cases} 1, & \text{if } c_j = \arg \max_{c_l \in \mathcal{C}} \left(\sum_{t_i \in \mathcal{T}} w_{d_k, t_i} f_{t_i, c_l}\right). \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

After updating the classes for each target word, the values of f_{t_i, c_j} are modified. This update is controlled by the correction rate η :

$$f_{t_i, c_j}^{(n+1)} = f_{t_i, c_j}^{(n)} + \eta \sum_{d_k \in \mathcal{D}} w_{d_k, t_i} \epsilon_{d_k, c_j}^{(n)}, \quad (6)$$

where the superscript (n) in f and ϵ denotes the value of these quantities computed in the n -th iteration of the algorithm and \mathcal{D} is the set of target words. Note that $\epsilon_{d_k, c_j}^{(n)}$ is computed as defined in equation 3. The values of ϵ_{d_k, c_j} and f_{t_i, c_j} in equations 3 and 6, respectively, are updated until a stop criterion is reached. In our experiments, we have stopped the algorithm when a minimum error $\epsilon_{min} = 0.01$ is obtained. If the minimum error is not reached after $n_{max} = 1,000$ iterations, the algorithm is stopped.

3. **Classification:** in the classification phase, the induced values of f are used in the classification. The word senses for each ambiguous word of the dataset are then obtained by computing the following linear combination:

$$\text{class}(d_k) = \arg \max_{c_j \in \mathcal{C}} \left(\sum_{t_i \in \mathcal{T}} w_{d_k, t_i} f_{t_i, c_j}\right). \quad (7)$$

4. Experimental Evaluation

This section presents the corpus used in the experiments. In addition, we also detail the experimental configuration of parameters. Finally, we present a robustness analysis to investigate how the performance of the IMBHN varies with the size of the training set.

4.1. Corpus

In order to evaluate the proposed algorithm, the SENSEVAL-2 [32] corpus was used. This corpus comprises documents from distinct sources, including the British National Corpus and the Penntreebank portion of the Wall Street Journal. The SENSEVAL-2 corpus encompasses 15,225 instances of short texts representing the context surrounding ambiguous words. Each word is tagged with its part-of-speech, and the manually annotated senses of four target words is provided. The number of senses and the number of instances of each word used in our experiments is shown in Table 1. In the evaluation process, these four words were considered as the target words. In particular, to characterized the contexts, we have removed stopwords and punctuation marks as such elements do not convey any semantical meaning and, therefore, do not improve the characterization of contexts.

Table 1: List of words used to evaluate the proposed word sense disambiguation algorithm. NS and NI denote the number of senses of the target word and the number of instances in the corpus, respectively. The dataset comprising word context and word senses was obtained from the SENSEVAL-2 corpus [32]. Prior to the application of the learning methods, stopwords and punctuation marks were removed from the original instances.

Target word	NS	NI
interest (noun)	6	2,368
line (noun)	6	4,146
serve (verb)	4	4,378
hard (adjective)	3	4,333

4.2. Experiment Configuration

The results obtained by the IMBHN algorithm were compared with four inductive classification algorithms: Naive Bayes (NB) [33], J48 (C4.5 algorithm) [34], IBk (k -Nearest Neighbors) [35] and Support Vector Machine via sequential minimal optimization (SMO) [36]. The parameters of these algorithms have been chosen using the methodology described in [37]. For the IMBHN algorithm, we used the error correction rates $\eta = \{0.01, 0.05, 0.10, 0.50\}$. The number of topical features used in the experiments were $|\mathcal{T}| = \{100, 200, 300\}$. Finally, the window size for the local features were $\omega = \{1, 2\}$. The evaluation process was performed via 10-fold cross-validation [38].

4.3. Results and discussion

To analyze the behavior and accuracy of the proposed algorithm, we first studied the WSD task using topical features to characterize the context of target words of our dataset. The obtained results are shown in Table 2. When the number of topical features $|\mathcal{T}|$ is set with $|\mathcal{T}| = 100$, the best results occurred for the SMO and J48 techniques. In three cases, the proposed algorithm IMBHN performed worse than the best results achieved with competing techniques.

In general, the performance of the classifiers tend to improve when the number of topical features ($|\mathcal{T}|$) increases from 100 to 300. This is clear when one observes that e.g. the best accuracy rate for the word “interest” goes from 79.77% to 84.71%. The same behavior can be observed for the other target words of the dataset, however, in a minor proportion. Concerning the performance of the proposed technique when $|\mathcal{T}| = \{200, 300\}$, in most cases, the IMBHN method is outperformed by the SMO technique, which provided the best results for the words “interest”, “line” and “serve”. The best results for the word “hard” was achieved with the J48 classifier.

When analyzing the performance of the classifiers induced with local features, a different pattern of accuracy has been found, as shown in Table 3. For the words “interest”, “line” and “serve” the IMBHN classifier yielded the best results, for $\omega = \{1, 2, 3\}$. Conversely, if we consider the word “hard”, the decision tree based algorithm, J48, outperformed all other

Table 2: Accuracy rates obtained by each algorithm using *topical* features to disambiguate the following target words: (i) “interest” (noun), (ii) “line” (noun), (iii) “serve” (verb) and (iv) “hard”. The best results for each value of $|\mathcal{T}|$ and for each target word are highlighted in bold font. The best results tend to occur with the SMO method, however, in particular cases, the J48 outperforms the SMO learning technique. Apart from the word “serve” when $|\mathcal{T}| = 300$, the IMBHN does not perform as good as the other traditional methods.

Method	$ \mathcal{T} $	interest	line	serve	hard
IMBHN	100	71.49%	59.91%	64.68%	77.28%
J48	100	79.47%	62.73%	68.15%	84.58%
IBk	100	75.71%	53.18%	63.68%	79.34%
NB	100	59.79%	51.95%	58.79%	43.04%
SMO	100	79.77%	62.87%	66.79%	84.07%
IMBHN	200	78.50%	65.53%	66.56%	78.74%
J48	200	82.39%	66.71%	68.95%	86.17%
IBk	200	80.70%	53.93%	63.24%	80.10%
NB	200	60.17%	54.43%	61.71%	42.69%
SMO	200	83.27%	68.95%	69.84%	85.36%
IMBHN	300	80.23%	67.82%	71.42%	78.62%
J48	300	82.68%	68.54%	70.67%	86.22%
IBk	300	80.32%	54.05%	63.13%	80.38%
NB	300	55.66%	54.14%	66.99%	41.61%
SMO	300	84.71%	69.87%	71.92%	85.52%
Baseline	–	52.80%	53.40%	41.40%	79.30%

methods. However, the performance achieved with J48 was very similar to the one obtained with the IMBHN: the maximum difference of accuracy between these two classifiers was 1.09%, when $\omega = 3$. This observation confirms the suitability of the proposed method for the problem, as optimized results have been found for virtually all words of the dataset.

The best results obtained with topical and local features are summarized in Table 4. The proposed algorithm for representing texts and discriminating senses outperformed other methods when considering also distinct types of features. In special, the IMBHN performed significantly better than the SMO method for the word “line” and “serve”. A minor gain in performance has been observed for “interest”. With regard to the word “hard”, the best performance was obtained with the J48 (with topical features). However, a similar accuracy was obtained with the IMBHN (with local features, as shown in Table 3). All in all, these results show, as a proof of principle, that the proposed algorithm may be useful to the word sense disambiguation problem, as optimal or near-optimal performance has been found in the studied corpus.

A disadvantage associated to the use of supervised methods to undertake the word sense disambiguation problem is the painstaking, time-consuming effort required to build reliable datasets [2]. For this reason, it becomes relevant to analyze the performance of WSD systems when only a few labelled instances are available for training [2]. In this sense, we performed a robustness analysis of the proposed algorithm to investigate how performance is affected when smaller fractions of the dataset are provided for the algorithm. To perform such a robustness analysis the following procedure was adopted. We defined a sampling rate \mathcal{S} , representing the percentage of *disregarded* instances from the original dataset. For each sampling rate, we computed the accuracy $\Gamma(S)$ relative to the sampled dataset. The relative accuracy rate for a given S was computed as

$$\tilde{\Gamma}(S) = \frac{\Gamma(S)}{\Gamma(0)}, \quad (8)$$

which quantifies the percentage of the original accuracy which is preserved when the original dataset is sampled with sampling rate S . For each sampling rate, we generated 50 sampled subsets. The obtained results for the IMBHN in its best configuration (i.e. using local

Table 3: Accuracy rates obtained by each algorithm using *local* features to disambiguate the following target words: (i) “interest” (noun), (ii) “line” (noun), (iii) “serve” (verb) and (iv) “hard”. The best results for each value of ω and for each target word are highlighted in bold font. For the words “interest”, “line” and “serve”, the best performance is achieved with the IMBHN method in all of the studied scenarios. For the word “hard”, the J48 learning algorithm displayed the best performance. However, in this case, the IMBHN method performed almost as well as the J48, for $\omega = \{1, 2, 3\}$. Another interesting pattern arising from the results is the fact that performances are improved when ω takes higher values.

Method	ω	interest	line	serve	hard
IMBHN	1	81.50%	69.19%	69.96%	85.50%
J48	1	65.83%	60.97%	46.43%	85.57%
IBk	1	74.73%	59.76%	62.54%	82.06%
NB	1	64.90%	37.16%	42.11%	43.94%
SMO	1	66.00%	62.61	57.88%	81.30%
IMBHN	2	83.27%	75.80%	78.48%	84.67%
J48	2	71.74%	61.21%	55.57%	85.39%
IBk	2	65.32%	56.72%	58.26%	78.35%
NB	2	66.97%	45.22%	60.16%	43.68%
SMO	2	64.10%	62.13	58.63%	80.68%
IMBHN	3	85.55%	77.13%	80.12%	84.16%
J48	3	76.85%	62.66%	60.94%	85.25%
IBk	3	52.44%	53.59%	52.12%	78.86%
NB	3	68.49%	50.43%	66.05%	42.97%
SMO	3	64.14%	60.80	58.45%	79.78%
Baseline	—	52.80%	53.40%	41.40%	79.30%

Table 4: Best classifiers for each feature set and its accuracy.

Target word	Topical features	Local features
interest (noun)	84.71% (SMO)	85.55% (IMBHN)
line (noun)	69.87% (SMO)	77.13% (IMBHN)
serve (verb)	71.92% (SMO)	80.12% (IMBHN)
hard (adjective)	86.22% (J48)	85.57% (J48)

features and $\omega = 3$) are shown in Figure 2. The best scenario occurs for the word “hard”, as even when 90% of the original is ignored, in average, more than 95% of the original accuracy (i.e. $\Gamma(S = 0)$) is recovered. Concerning the other words, a good performance was also observed when only a small fraction was available. This is the case of “serve”: when 90% of the dataset is disregarded, 85% of the original accuracy is kept. These results suggest that the IMBHN could be successfully applied in much smaller datasets without a significative loss in performance. We have found similar robustness results for other configurations of parameters (ω) of the IMBHN (results not shown), which reinforces the hypothesis that the resiliency of the method with regard to the total amount of instances in the training phase is stable with varying parameter values. Note that such a robustness, although strongly desired in practical problems, does not naturally arise in all pattern recognition methods. This is evident e.g. when the robustness SMO is verified for “serve” and “interest”, as shown in Figure 3. Note that when $S = 0.9$, the accuracy drops to about 60% of its original value.

5. Conclusion

The accurate discrimination of word senses plays a pivotal role in information extraction and document classification tasks. While methods based on deep paradigms may perform well in very specific domains, statistical methods based mainly on machine learning have proved useful to undertake the word sense disambiguation task in more general contexts. In this article, we have devised a statistical model to both represent contexts and recognize patterns in written texts. The model hinges on a bipartite network, with layers representing

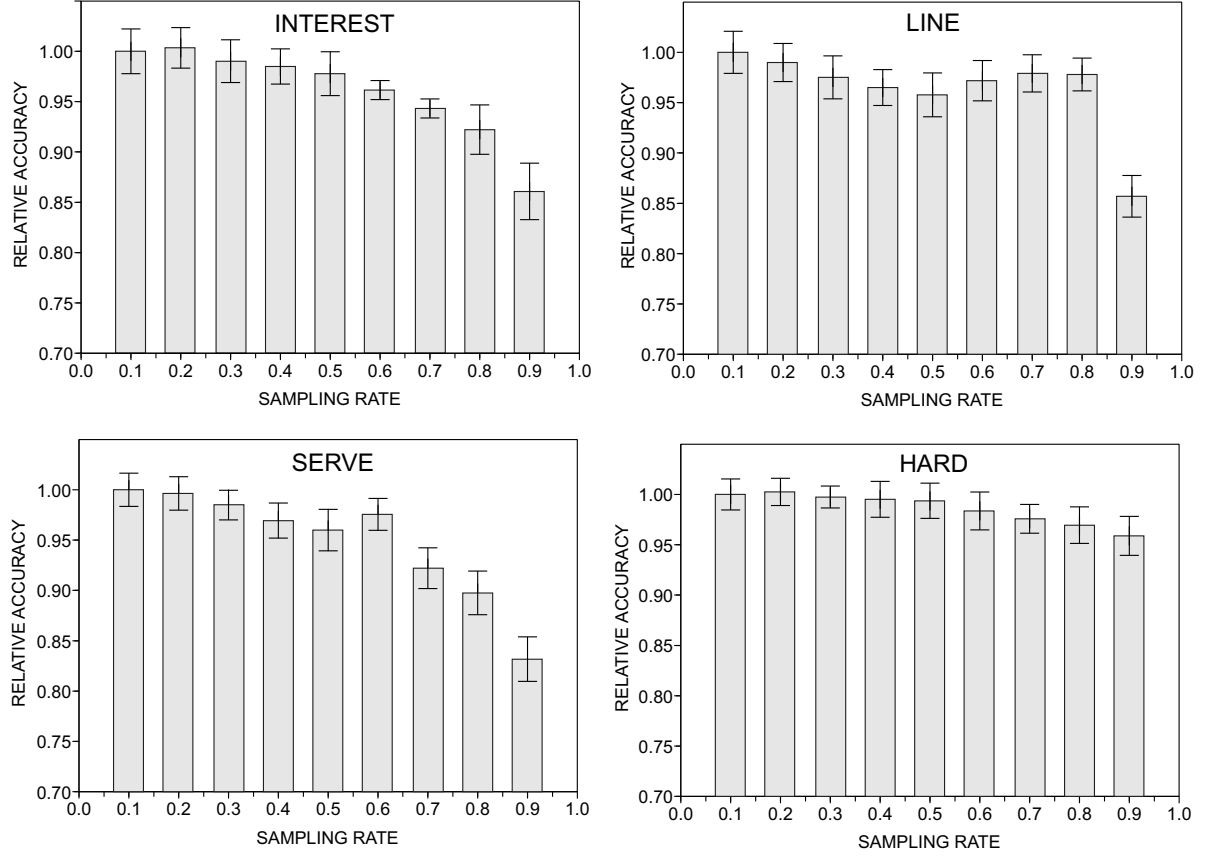


Figure 2: Robustness analysis performed with the IMBHN algorithm. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by equation 8. Note that, in the worst case, the accuracy of the IMBHN reaches 85% of the accuracy when only 10% of the original data is available ($S = 0.9$), confirming thus the robustness of the method. A similar behavior was obtained when the approach based on topical features was evaluated with $\omega = \{1, 2\}$.

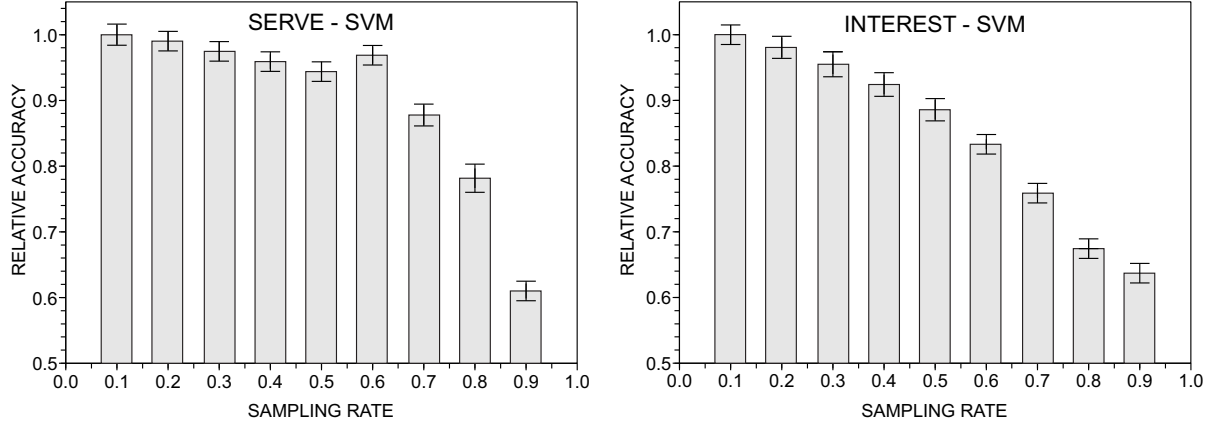


Figure 3: Robustness analysis performed with the SMO algorithm for two words of the dataset. The sampling rate corresponds to the fraction (percentage) of instances randomly removed from the original dataset. The relative accuracy is given by equation 8. Unlike the IMBHN algorithm, the accuracy rate drops significantly for high sampling rates.

feature words and target words, i.e. words conveying two or more potential senses. We have shown, as a proof of principle, that the proposed model presents a significant performance, mainly when contextual features are modelled via extraction of local words to represent semantical contexts. We have also observed that, in general, our method performs well even if a relatively small amount of data is available for the training process. This is an important property as it may significantly reduce both time and effort required to construct a corpus of labelled data.

As future works, we intend to explore further generalizations of the algorithm. Owing to the power of word adjacency networks in extracting relevant semantical features of texts [15], we intend to use such models to improve the characterization of the studied bipartite networks. The word adjacency model could be used, for example, to better represent the relationship between feature and target words by using network similarity measurements [39, 40, 41]. We also intend to extend the present model to consider topological and dynamical measurements of word adjacency networks as local features [15].

Acknowledgements

E.A.C. Jr. and D.R.A. acknowledge financial support from Google (Google Research Awards in Latin America grant). D.R.A. thanks São Paulo Research Foundation (FAPESP) for support (grant. no. 2014/20830-0). A.A.L. acknowledges support from FAPESP (grant no. 2011/22749-8 and 2015/14228-9) and CNPq (Brazil) (grant no. 302645/2015-2).

References

- [1] C. D. Manning, H. Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA, USA, 1999.
- [2] R. Navigli, Word sense disambiguation: A survey, ACM Computing Surveys (CSUR) 41 (2009) 10.
- [3] J. C. Mallery, Thinking about foreign policy: Finding an appropriate role for artificially intelligent computers, in: Master's thesis, MIT Political Science Department, Citeseer.
- [4] W. Weaver, Translation, Machine translation of languages 14 (1955) 15–23.
- [5] K. W. Church, L. F. Rau, Commercial applications of natural language processing, Communications of the ACM 38 (1995) 71–79.
- [6] C. Stokoe, M. P. Oakes, J. Tait, Word sense disambiguation in information retrieval revisited, in: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval, ACM, pp. 159–166.
- [7] K. Markert, M. Nissim, Semeval-2007 task 08: Metonymy resolution at semeval-2007, in: Proceedings of the 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics, pp. 36–41.
- [8] X. Zhou, H. Han, Survey of word sense disambiguation approaches., in: FLAIRS Conference, pp. 307–313.
- [9] D. Fernandez-Amoros, R. Heradio, Understanding the role of conceptual relations in word sense disambiguation, Expert Systems with Applications 38 (2011) 9506 – 9516.
- [10] D. Spina, J. Gonzalo, E. Amigó, Discovering filter keywords for company name disambiguation in twitter, Expert Systems with Applications 40 (2013) 4986 – 5003.
- [11] N. Fernandez, J. A. Fisteus, L. Sanchez, G. Lopez, Identity rank: Named entity disambiguation in the news domain, Expert Systems with Applications 39 (2012) 9207 – 9221.
- [12] T. Berners-Lee, J. Hendler, O. Lassila, et al., The semantic web, Scientific american 284 (2001) 28–37.
- [13] F. Breve, L. Zhao, M. Quiles, W. Pedrycz, J. Liu, Particle competition and cooperation in networks for semi-supervised learning, IEEE Transactions on Knowledge and Data Engineering 24 (2012) 1686–1698.
- [14] J. Machicao, A. G. Marco, O. M. Bruno, Chaotic encryption method based on life-like cellular automata, Expert Systems with Applications 39 (2012) 12626 – 12635.
- [15] D. R. Amancio, O. N. Oliveira Jr, L. d. F. Costa, Unveiling the relationship between complex networks metrics and word senses, EPL (Europhysics Letters) 98 (2012) 18002.
- [16] R. Mihalcea, D. Radev, Graph-based natural language processing and information retrieval, Cambridge University Press, 2011.
- [17] J. Véronis, Hyperlex: lexical cartography for information retrieval, Computer Speech & Language 18 (2004) 223–252.

- [18] D. R. Amancio, E. G. Altmann, D. Rybski, O. N. Oliveira, Jr, L. d. F. Costa, Probing the statistical properties of unknown texts: application to the voynich manuscript, *PLoS ONE* 8 (2013) 1–10.
- [19] S. Segarra, M. Eisen, A. Ribeiro, Authorship attribution through function word adjacency networks, *IEEE Transactions on Signal Processing* 63 (2015) 5464–5478.
- [20] H. Liu, The complexity of chinese syntactic dependency networks, *Physica A* 387 (2008) 3048 – 3058.
- [21] A. P. Masucci, G. J. Rodgers, Network properties of written human language, *Phys. Rev. E* 74 (2006) 026102.
- [22] T. C. Silva, D. R. Amancio, Word sense disambiguation via high order of learning in complex networks, *EPL (Europhysics Letters)* 98 (2012) 58001.
- [23] T. C. Silva, D. R. Amancio, Discriminating word senses with tourist walks in complex networks, *Eur. Phys. J. B* 86 (2013) 297.
- [24] R. J. Mooney, Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning, *arXiv preprint cmp-lg/9612001* (1996).
- [25] G. Escudero, L. Màrquez, G. Rigau, J. G. Salgado, On the portability and tuning of supervised word sense disambiguation systems (2000).
- [26] Y. K. Lee, H. T. Ng, An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation, in: *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, Association for Computational Linguistics, pp. 41–48.
- [27] G. A. Wachs-Lopes, P. S. Rodrigues, Analyzing natural human language from the point of view of dynamic of a complex network, *Expert Systems with Applications* 45 (2016) 8 – 22.
- [28] D. R. Amancio, A complex network approach to stylometry, *PLoS ONE* 10 (2015) e0136076.
- [29] D. R. Amancio, S. M. Aluisio, O. N. Oliveira Jr., L. da F. Costa, Complex networks analysis of language complexity, *EPL (Europhysics Letters)* 100 (2012) 58002.
- [30] K. Sneppen, M. Rosvall, A. Trusina, P. Minnhagen, A simple model for self-organization of bipartite networks, *EPL (Europhysics Letters)* 67 (2004) 349.
- [31] R. G. Rossi, A. de Andrade Lopes, T. de Paulo Faleiros, S. O. Rezende, Inductive model generation for text classification using a bipartite heterogeneous network, *Journal of Computer Science and Technology* 29 (2014) 361–375.
- [32] P. Edmonds, S. Cotton, Senseval-2: overview, in: *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, Association for Computational Linguistics, pp. 1–5.
- [33] R. Caruana, A. Niculescu-Mizil, An empirical comparison of supervised learning algorithms, in: *Proceedings of the 23rd International Conference on Machine Learning, ICML '06*, ACM, New York, NY, USA, 2006, pp. 161–168.

- [34] R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
- [35] D. W. Aha, D. Kibler, M. K. Albert, Instance-based learning algorithms, *Mach. Learn.* 6 (1991) 37–66.
- [36] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C. Burges, A. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.
- [37] D. R. Amancio, C. H. Comin, D. Casanova, G. Travieso, O. M. Bruno, F. A. Rodrigues, L. d. F. Costa, A systematic comparison of supervised classifiers, *PLoS ONE* 9 (2014) e94137.
- [38] R. Kohavi, A study of cross-validation and bootstrap for accuracy estimation and model selection, in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI'95*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995, pp. 1137–1143.
- [39] J.-G. Liu, L. Hou, X. Pan, Q. Guo, T. Zhou, Stability of similarity measurements for bipartite networks, *Scientific Reports* 6 (2016) 18653 EP.
- [40] C. H. Comin, T. K. D. Peron, F. N. Silva, D. R. Amancio, F. A. Rodrigues, L. F. Costa, Complex systems: features, similarity and connectivity, *arXiv: 1606.05400* (2016) 1–61.
- [41] E. A. Leicht, P. Holme, M. E. J. Newman, Vertex similarity in networks, *Phys. Rev. E* 73 (2006) 026120.